

## Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application.

### Listing of Claims:

1 1-47. (Cancelled).

1 48. (Currently amended) A method implemented by a secondary server to maintain a  
2 secondary directory, comprising:  
3 receiving notification from a primary server that a particular ~~update~~ non-idempotent  
4 operation has been ~~made performed~~ by the primary server ~~to on~~ a primary  
5 directory;  
6 in response to the notification, ~~making~~ performing the particular ~~update~~ non-  
7 idempotent operation ~~to on~~ the secondary directory such that the ~~second~~  
8 secondary directory mirrors the primary directory;  
9 receiving a request, from a client, to ~~make perform~~ the particular ~~update~~ non-  
10 idempotent operation ~~to on~~ the secondary directory, wherein the client sends  
11 the request because the client has not received confirmation that the primary  
12 server has ~~made performed~~ the particular ~~update~~ non-idempotent operation ~~to~~  
13 on the primary directory;  
14 ~~determining~~ recognizing that the particular ~~update~~ non-idempotent operation,  
15 identified in the request, cannot be ~~made performed to on~~ the secondary  
16 directory because the particular ~~update~~ non-idempotent operation has already  
17 been ~~made performed to on~~ the secondary directory as a result of receiving  
18 the notification from the primary server; and

19 in response to the ~~determination~~ recognition that the particular ~~update non-~~  
20 idempotent operation has already been ~~made performed to~~ on the secondary  
21 directory, the secondary server sending an indication to the client that the  
22 particular ~~update non-idempotent operation~~ was successfully ~~made to~~  
23 performed on the secondary directory even though the non-idempotent  
24 operation was not performed on the secondary directory after receiving the  
25 request.

1 49. (Currently amended) The method of Claim 48, wherein said particular ~~update non-~~  
2 idempotent operation ~~involves is~~ is an add operation.

1 50. (Currently amended) The method of Claim 48, wherein said particular ~~update non-~~  
2 idempotent operation ~~involves is~~ is a delete operation.

1 51. (Currently amended) The method of Claim 48, wherein said particular ~~update non-~~  
2 idempotent operation ~~involves is~~ is adding a rename operation.

1 52. (Currently amended) A machine-readable medium carrying one or more sequences  
2 of instructions for maintaining a secondary directory, wherein execution of the one  
3 or more sequences of instructions by one or more processors of a secondary server  
4 causes the one or more processors to perform the steps of:  
5 receiving notification from a primary server that a particular ~~update non-idempotent~~  
6 operation has been ~~made performed~~ by the primary server ~~to on~~ on a primary  
7 directory;  
8 in response to the notification, ~~making performing~~ the particular ~~update non-~~  
9 idempotent operation ~~to on~~ on the secondary directory such that the ~~second~~  
10 secondary directory mirrors the primary directory;

11 receiving a request, from a client, to ~~make~~ perform the particular ~~update~~ non-  
12 idempotent operation ~~to~~ on the secondary directory, wherein the client sends  
13 the request because the client has not received confirmation that the primary  
14 server has ~~made~~ performed the particular ~~update~~ non-idempotent operation ~~to~~  
15 on the primary directory;  
16 ~~determining~~ recognizing that the particular ~~update~~ non-idempotent operation,  
17 identified in the request, cannot be ~~made~~ performed ~~to~~ on the secondary  
18 directory because the particular ~~update~~ non-idempotent operation has already  
19 been ~~made~~ performed ~~to~~ on the secondary directory as a result of receiving  
20 the notification from the primary server; and  
21 in response to the ~~determination~~ recognition that the particular ~~update~~ non-  
22 idempotent operation has already been ~~made~~ performed ~~to~~ on the secondary  
23 directory, the secondary server sending an indication to the client that the  
24 particular ~~update~~ non-idempotent operation was successfully ~~made~~ performed  
25 ~~to~~ on the secondary directory even though the non-idempotent operation was  
26 not performed on the secondary directory after receiving the request.

1 53. (Currently amended) The machine-readable medium of Claim 52, wherein said  
2 particular ~~update~~ non-idempotent operation ~~involves~~ is an add operation.

1 54. (Currently amended) The machine-readable medium of Claim 52, wherein said  
2 particular ~~update~~ non-idempotent operation ~~involves~~ is a delete operation.

1 55. (Currently amended) The machine-readable medium of Claim 52, wherein said  
2 particular ~~update~~ non-idempotent operation ~~involves~~ is adding a rename operation.

1 56. (Currently amended) An apparatus for maintaining a secondary directory,  
2 comprising:  
3 a ~~module~~ means for receiving notification from a primary server that a particular  
4 ~~update non-idempotent operation~~ has been ~~made~~ performed by the primary  
5 server ~~to~~ on a primary directory;  
6 a ~~module~~ means for ~~making performing~~ the particular ~~update non-idempotent~~  
7 ~~operation~~ to on a secondary directory such that the ~~second~~ secondary  
8 directory mirrors the primary directory in response to receiving the  
9 notification;  
10 a ~~module~~ means for receiving a request, from a client, to ~~make~~ perform the particular  
11 ~~update non-idempotent operation~~ to on the secondary directory, wherein the  
12 client sends the request because the client has not received confirmation that  
13 the primary server has ~~made performed~~ the particular ~~update non-idempotent~~  
14 ~~operation~~ to on the primary directory;  
15 a ~~module~~ means for ~~determining~~ recognizing that the particular ~~update non-~~  
16 ~~idempotent operation, identified in the request, cannot be~~ made performed to  
17 on the secondary directory because the particular ~~update non-idempotent~~  
18 ~~operation~~ has already been ~~made performed~~ to on the secondary directory as  
19 a result of receiving the notification from the primary server; and  
20 a ~~module~~ means for sending, from the secondary server, an indication to the client  
21 that the particular ~~update non-idempotent operation~~ was successfully ~~made~~  
22 ~~performed~~ to on the secondary directory in response to ~~determining the~~  
23 ~~recognition~~ that the particular ~~update non-idempotent operation~~ has already  
24 been ~~made performed~~ to on the secondary directory even though the non-

25                    idempotent operation was not performed on the secondary directory after  
26                    receiving the request.

1    57.    (Currently amended) The apparatus of Claim 56, wherein said particular ~~update~~ non-  
2           idempotent operation ~~involves~~ is an add operation.

1    58.    (Currently amended) The apparatus of Claim 56, wherein said particular ~~update~~ non-  
2           idempotent operation ~~involves~~ is a delete operation.

1    59.    (Currently amended) The apparatus of Claim 56, wherein said particular ~~update~~ non-  
2           idempotent operation ~~involves~~ is adding a rename operation.

1    60.    (New) The method of Claim 48, wherein said primary directory and said secondary  
2           directory are both LDAP directories.

1    61.    (New) The machine-readable medium of Claim 52, wherein said primary directory  
2           and said secondary directory are both LDAP directories.

1    62.    (New) The apparatus of Claim 56, wherein said primary directory and said  
2           secondary directory are both LDAP directories.